

Diversity Embedded Streaming Erasure Codes (DE-SCo): Constructions and Optimality

Ahmed Badr and Ashish Khisti
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, ON, M5S 3G4, Canada
Email: {abadr, akhisti}@comm.utoronto.ca

Emin Martinian
Signals, Information and Algorithms Laboratory
Massachusetts Inst. of Technology
Cambridge, MA, 02139
Email: emin@alum.mit.edu

Abstract—Streaming erasure codes encode a source stream to guarantee that each source packet is recovered within a fixed delay at the receiver over a burst-erasure channel. This paper introduces *diversity embedded streaming erasure codes* (DE-SCo), that provide a flexible tradeoff between the channel quality and receiver delay. When the channel conditions are good, the source stream is recovered with a low delay, whereas when the channel conditions are poor the source stream is still recovered, albeit with a larger delay. Information theoretic analysis of the underlying burst-erasure broadcast channel reveals that DE-SCo achieve the minimum possible delay for the weaker user, without sacrificing the performance of the stronger user. A larger class of multicast streaming erasure codes (MU-SCo) that achieve optimal tradeoff between rate, delay and erasure-burst length is also constructed.

I. INTRODUCTION

Forward error correction codes designed for streaming sources require that (a) the channel input stream be produced sequentially from the source stream (b) the decoder sequentially reconstructs the source stream as it observes the channel output. In contrast, traditional error correction codes such as maximum distance separable (MDS) codes map blocks of data to a codeword and the decoder waits until the entire codeword is received before the source data can be reproduced. Rateless codes such as the digital fountain codes also do not form ideal streaming codes. First they require that the entire source data be available before the output stream is reproduced. Secondly they provide no guarantees on the sequential reconstruction of the source stream. Non-block codes such as convolutional codes in conjunction with sequential decoding can be designed for low delay applications [4]. However to our best knowledge, these constructions need to be optimized through a numerical search for finite constraint lengths. Low-delay codes with feedback are recently studied in [5], [6] while compression of streaming sources is studied in [7].

In [1, Chapter 8] a new class of codes, *streaming erasure codes* (SCo) are proposed. The encoder observes a semi-infinite source stream — one packet is revealed in each time slot — and maps it to a coded output stream of rate R . The channel is modelled as a burst-erasure channel. Starting at an arbitrary time, it introduces an erasure-burst of maximum length B . The decoder is required to reconstruct each source packet with a maximum delay T . A fundamental relationship between R , B and T is established and SCo codes are

constructed that achieve this tradeoff. We emphasize that the parity check symbols in these constructions involve a careful combination of source symbols. In particular, random linear combinations, popularly used in e.g., network coding, do not attain the optimal performance.

The SCo framework however requires that the value of B and T be known apriori. In practice this forces a conservative design i.e., we design the code for the worst case B thereby incurring a higher overhead (or a larger delay) even when the channel is relatively good. Moreover there is often a flexibility in the allowable delay. Techniques such as adaptive media playback [8] have been designed to tune the play-out rate as a function of the received buffer size to deal with a temporary increase in delay. Hence it is not desirable to have to fix T during the design stage either.

The streaming codes introduced in this work do not commit apriori to a specific delay. Instead they realize a delay that depends on the channel conditions. At an information theoretic level, our setup extends the point-to-point link in [1] to a multicast model — there is one source stream and two receivers. The channel for each receiver introduces an erasure-burst of length B_i and each receiver can tolerate a delay of T_i . We investigate multicast streaming erasure code (MU-SCo) constructions that achieve the maximum rate under these constraints. We primarily focus on a particular subclass — diversity embedded streaming erasure codes (DE-SCo). These codes modify a single user SCo such that the resulting code can support a second user, whose channel introduces a larger erasure-burst, without sacrificing the performance of the first user. Our construction embeds new parity checks in an SCo code in a manner such that (a) no interference is caused to the stronger (and low delay) user and (b) the weaker user can use some of the parity checks of the stronger user as side information to recover part of the source symbols. DE-SCo constructions outperform baseline schemes that simply concatenate the single user SCo for the two users. An information theoretic converse establishes that DE-SCo achieves the minimum possible delay for the weaker receiver without sacrificing the performance of the stronger user. Finally all our code constructions can be encoded and decoded with a polynomial time complexity in T and B .

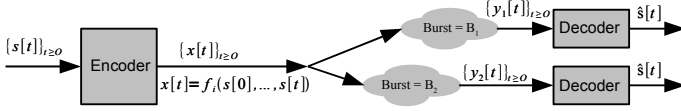


Fig. 1: The source stream $\{s[i]\}$ is causally mapped into an output stream $\{x[i]\}$. Both the receivers observe these packets via their channels. The channel introduces an erasure-burst of length B_i , and each receiver tolerates a delay of T_i , for $i = 1, 2$.

II. SYSTEM MODEL

The transmitter encodes a stream of source packets $\{s[t]\}_{t \geq 0}$ intended to be received at two receivers as shown in Fig. 1. The channel packets $\{x[t]\}_{t \geq 0}$ are produced causally from the source stream, $x[t] = f_t(s[0], \dots, s[t])$.

The channel of receiver i introduces an erasure-burst of length B_i i.e., the channel output at receiver i at time t is given by

$$y_i[t] = \begin{cases} \star & t \in [j_i, j_i + B_i - 1] \\ x[t] & \text{otherwise} \end{cases} \quad (1)$$

for $i = 1, 2$ and for some $j_i \geq 0$. Furthermore, user i tolerates a delay of T_i , i.e., there exists a sequence of decoding functions $\gamma_{1t}(\cdot)$ and $\gamma_{2t}(\cdot)$ such that

$$\hat{s}_i[t] = \gamma_{it}(y_1[0], y_1[1], \dots, y_1[i + T_i]), \quad i = 1, 2, \quad (2)$$

and $\Pr(s_i[t] \neq \hat{s}_i[t]) = 0, \quad \forall t \geq 0, \quad .$

The source stream is an i.i.d. process; each source symbol is sampled from a distribution $p_s(\cdot)$. The rate of the multicast code is defined as ratio of the (marginal) entropy of the source symbol to the (marginal) entropy of each channel symbol i.e., $R = H(s)/H(x)$. An *optimal multicast streaming erasure code (MU-SCo)* achieves the maximum rate for a given choice of (B_i, T_i) . Of particular interest is the following subclass.

Definition 1 (Diversity Embedded Streaming Erasure Codes (DE-SCo)). *Consider the multicast model in Fig. 1 where the channels of the two receivers introduce an erasure burst of lengths B_1 and B_2 respectively with $B_1 < B_2$. A DE-SCo is a rate $R = \frac{T_1}{T_1 + B_1}$ MU-SCo construction that achieves a delay T_1 at receiver 1 and supports receiver 2 with delay T_2 . An optimal DE-SCo minimizes the delay T_2 at receiver 2 for given values of B_1, T_1 and B_2 .*

Our setup generalizes the point-to-point case in [1, Chapter 8] where single user SCo codes for parameters (B_1, T_1) achieve the streaming capacity $\frac{T_1}{T_1 + B_1}$. An optimal MU-SCo construction, attains the maximum rate for fixed parameters (B_i, T_i) . An optimal DE-SCo fixes the rate to the capacity of user 1, and supports user 2 with the minimum possible delay T_2 .

Note that our model only considers a single erasure burst on each channel. As is the case with (single user) SCo, our constructions correct multiple erasure-bursts separated sufficiently apart. Also we only consider the erasure channel model. It naturally arises when these codes are implemented in application layer multimedia encoding. More general channel

models can be transformed into an erasure model by applying an appropriate inner code [1, Chapter 7].

III. EXAMPLE

We first highlight our results via a numerical example: $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 4)$. Single user SCo constructions from [1], [2] for both users are illustrated in Table I(a) and I(b) respectively. In each case, the source symbol $s[i]$ is split into two sub-symbols $(s_0[i], s_1[i])$ and the channel symbol $x[i]$ is obtained by concatenating the source symbol $s[i]$ with a parity check symbol $p[i]$. In the $(1, 2)$ SCo construction, parity check symbol $p_A[i] = s_1[i-1] \oplus s_0[i-2]$ is generated by combining the source sub-symbols diagonally across the source stream as illustrated with the rectangular boxes. For the $(B, T) = (2, 4)$, the choice $p_B[i] = s_1[i-2] \oplus s_0[i-4]$ is similar, except that an interleaving of step of size 2 is applied before the parity checks are produced.

How can we construct a single code that simultaneously supports both $\{(1, 2), (2, 4)\}$? The first approach is to concatenate the two parity check streams as shown in Table. I(c). Each receiver ignores the parity check rows of the other receiver and performs single user decoding. However such a concatenated streaming code (Cc-SCo) achieves a rate of $1/2$, whereas the optimal MU-SCo construction achieves a rate of $3/5$. Before specifying the optimal MU-SCo code construction, let us first consider designing a DE-SCo (see Def. 1) for this setup. Recall that this code, of rate $2/3$, satisfies $(B_1, T_1) = (1, 2)$, also achieves minimum delay T_2 for $B_2 = 2$.

In Table II(a) we illustrate a rate $R = 2/3$ code that achieves $T_2 = 6$. It is obtained by shifting the parity checks of the SCo code in Table I(b) to the right by two symbols and combining with the parity checks of the SCo code in Table I(a) i.e., $q[i] = p_A[i] \oplus p_B[i-2]$. Note that parity check symbols $p_B[\cdot]$ do not interfere with the parity checks of user 1 i.e., when $s[i]$ is erased, receiver 1 can recover $p_A[i+1]$ and $p_A[i+2]$ from $q[i+1]$ and $q[i+2]$ respectively by cancelling $p_B[\cdot]$ that combine with these symbols. It then recovers $s[i]$. Likewise if $s[i]$ and $s[i-1]$ are erased, then receiver 2 recovers $p_B[i+1], \dots, p_B[i+4]$ from $q[i+3], \dots, q[i+6]$ respectively by cancelling out the interfering $p_A[\cdot]$, thus yielding $T_2 = 6$.

The interference avoidance strategy illustrated above is sub-optimal. Table. II(b) shows the DE-SCo construction that achieves the minimum possible delay of $T_2 = 5$. In this construction we first construct the parity checks $\check{p}_B[i] = s_1[i-2] \oplus s_0[i-1]$ by combining the source symbols along the opposite diagonal of the $(1, 2)$ SCo code in Table I(a). Note that $x(i) = (s[i], \check{p}_B[i])$ is also a single user $(1, 2)$ SCo code. We then shift the parity check stream to the right by $T+B = 3$ symbols and combine with $p_A[i]$ i.e., $q[i] = p_A[i] \oplus \check{p}_B[i-3]$. In the resulting code, receiver 1 is still able to cancel the effect of $\check{p}_B[\cdot]$ as before and achieve $T_1 = 2$. Furthermore at receiver 2 if $s[i]$ and $s[i-1]$ are erased, then observe that receiver 2 obtains $s_0[i]$ and $s_0[i-1]$ from $q[i+2]$ and $q[i+3]$ respectively and $s_1[i-1]$ and $s_1[i]$ from $q[i+4]$ and $q[i+5]$ respectively, thus yielding $T_2 = 5$ symbols.

(a) SCo Construction for $(B, T) = (1, 2)$

$s_0[i-1]$ $s_1[i-1]$	$s_0[i]$ $s_1[i]$	$s_0[i+1]$ $s_1[i+1]$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$s_0[i-3] \oplus s_1[i-2]$	$s_0[i-2] \oplus s_1[i-1]$	$s_0[i-1] \oplus s_1[i]$	$s_0[i] \oplus s_1[i+1]$	$s_0[i+1] \oplus s_1[i+2]$	$s_0[i+2] \oplus s_1[i+3]$

(b) SCo Construction for $(B, T) = (2, 4)$

$s_0[i-1]$ $s_1[i-1]$	$s_0[i]$ $s_1[i]$	$s_0[i+1]$ $s_1[i+1]$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$s_0[i-5] \oplus s_1[i-3]$	$s_0[i-4] \oplus s_1[i-2]$	$s_0[i-3] \oplus s_1[i-1]$	$s_0[i-2] \oplus s_1[i]$	$s_0[i-1] \oplus s_1[i+1]$	$s_0[i] \oplus s_1[i+2]$

(c) Cc-SCo for $\{(B_1, T_1), (B_2, T_2)\} = \{(1, 2), (2, 4)\}$

$s_0[i-1]$ $s_1[i-1]$	$s_0[i]$ $s_1[i]$	$s_0[i+1]$ $s_1[i+1]$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$s_0[i-3] \oplus s_1[i-2]$	$s_0[i-2] \oplus s_1[i-1]$	$s_0[i-1] \oplus s_1[i]$	$s_0[i] \oplus s_1[i+1]$	$s_0[i+1] \oplus s_1[i+2]$	$s_0[i+2] \oplus s_1[i+3]$
$s_0[i-5] \oplus s_1[i-3]$	$s_0[i-4] \oplus s_1[i-2]$	$s_0[i-3] \oplus s_1[i-1]$	$s_0[i-2] \oplus s_1[i]$	$s_0[i-1] \oplus s_1[i+1]$	$s_0[i] \oplus s_1[i+2]$

TABLE I: Single user SCo constructions are shown in the upper two figures. The Cc-SCo construction in the last figure repeats the parity checks of the two single user codes to simultaneously satisfy both the users. This baseline approach is in general sub-optimal

(a) IA-SCo Code Construction for $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 6)$

$s_0[i-1]$ $s_1[i-1]$	$s_0[i]$ $s_1[i]$	$s_0[i+1]$ $s_1[i+1]$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$s_0[i-3] \oplus s_1[i-2]$ \oplus $s_0[i-7] \oplus s_1[i-5]$	$s_0[i-2] \oplus s_1[i-1]$ \oplus $s_0[i-6] \oplus s_1[i-4]$	$s_0[i-1] \oplus s_1[i]$ \oplus $s_0[i-5] \oplus s_1[i-3]$	$s_0[i] \oplus s_1[i+1]$ \oplus $s_0[i-4] \oplus s_1[i-2]$	$s_0[i+1] \oplus s_1[i+2]$ \oplus $s_0[i-3] \oplus s_1[i-1]$	$s_0[i+2] \oplus s_1[i+3]$ \oplus $s_0[i-2] \oplus s_1[i]$

(b) DE-SCo Code Construction for $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 5)$

$s_0[i-1]$ $s_1[i-1]$	$s_0[i]$ $s_1[i]$	$s_0[i+1]$ $s_1[i+1]$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$\{s_0[i-3] \oplus s_1[i-2]$ \oplus $s_1[i-6] \oplus s_0[i-5]\}$	$\{s_0[i-2] \oplus s_1[i-1]$ \oplus $s_1[i-5] \oplus s_0[i-4]\}$	$\{s_0[i-1] \oplus s_1[i]$ \oplus $s_1[i-4] \oplus s_0[i-3]\}$	$\{s_0[i] \oplus s_1[i+1]$ \oplus $s_1[i-3] \oplus s_0[i-2]\}$	$\{s_0[i+1] \oplus s_1[i+2]$ \oplus $s_1[i-2] \oplus s_0[i-1]\}$	$\{s_0[i+2] \oplus s_1[i+3]$ \oplus $s_1[i-1] \oplus s_0[i]\}$

TABLE II: Rate 2/3 code constructions that satisfy user 1 with $(B_1, T_1) = (1, 2)$ and user 2 with $B_2 = 2$.

Finally, the $R = 3/5$, MU-SCo $\{(1, 2), (2, 4)\}$ construction is as follows. Split each source symbol $s[i]$ into six sub-symbols $s_0[i], \dots, s_5[i]$ and construct an expanded source sequence $\tilde{s}[\cdot]$ such that $\tilde{s}[2i] = (s_0[i], s_1[i], s_2[i])$ and $\tilde{s}[2i+1] = (s_3[i], s_4[i], s_5[i])$. There exists a $\{(2, 3), (4, 8)\}$ DE-SCo code (see Thm. 1) that we apply to $\tilde{s}[\cdot]$ to produce the parity checks $\tilde{p}[\cdot]$ and transmit $p[i] = (\tilde{p}[2i], \tilde{p}[2i+1])$ along with $s[i]$ at time i . Due to the properties of the DE-SCo construction and our source expansion, the resulting code corrects a single erasure with a delay of 2 symbols and an erasure-burst of length 2 with a delay of 4. An explicit construction is provided in the full paper [9] due to space constraints.

IV. CONSTRUCTION OF DE-SCo

In this section we describe the DE-SCo construction. We rely on several properties of the single user SCo; see [1, Chapter 7-8],[2] for the background; [9] for a detailed proof.

Theorem 1. Let $(B_1, T_1) = (B, T)$ and suppose $B_2 = \alpha B$ where α is any integer that exceeds 1. The minimum possible delay for any code of rate $R = \frac{T}{T+B}$ is

$$T_2^* = \alpha T + B, \quad (3)$$

and is achieved by the optimal DE-SCo construction.

The converse in Theorem 1 states that the rate of any $\{(B, T), (B_2, T_2)\}$ MU-SCo with $T_2 < T_2^*$ is strictly below R . We establish this by constructing a periodic burst-erasure channel in which every period of $(\alpha - 1)B + T_2$ symbols consists of a sequence of αB erasures followed by a sequence of non-erased symbols. Following a similar line of reasoning as in [3, Theorem 2] we can show that a MU-SCo code corrects all erasures on this channel. The rate of the MU-SCo code then must not exceed $1 - \frac{\alpha B}{(\alpha - 1)B + T_2}$ which is less than R if $T_2 < T_2^*$.

For achievability, T_2^* in (3) we construct the following code:

parity checks marked by 1, 2, 3 and more generally all $\mathbf{p}^B[t]$ for $t \geq 7$. Note that this allows us to recover the non-urgent (i.e., the top three rows) erased sub-symbols in $\mathbf{d}_{-8}^B, \dots, \mathbf{d}_{-5}^B$ by $t = 9$. Next, compute parity checks marked by 1, 2, 3 of \mathcal{C}_2 in the shaded area in Fig. 2 and cancel them and recover the parity checks marked by a, b and c . At this point we have all the parity checks $\mathbf{p}^A[\cdot]$ for $0 \leq t \leq 6$. Since the diagonals $\mathbf{d}_{-1}^A, \dots, \mathbf{d}_{-4}^A$ involve four or fewer erasures we can now recover these sub-symbols.

The remaining non-urgent sub-symbols need to be recovered in a recursive manner. Note that \mathbf{d}_{-5}^A , marked by e , has five erased symbols. However the first symbol marked by $(4e)$ also belongs to \mathbf{d}_{-5}^B and has already been recovered. The remaining four sub-symbols can be recovered by the four available parity checks of $\mathcal{p}_A[\cdot]$ marked by e . Similarly \mathbf{d}_{-4}^B , marked by 5, also has five erasures, but the first symbol $(5d)$ also belongs to \mathbf{d}_{-4}^A and has been recovered. Hence the remaining parity checks can be recovered using the parity checks of $\mathbf{p}^B[\cdot]$. Of these, by construction of \mathcal{C}_2 , the non-urgent symbols will be recovered by time $\mathcal{T} = 9$. The decoder then recovers \mathbf{d}_{-6}^A and \mathbf{d}_{-3}^B in the next step to recover all non-urgent sub-symbols.

V. OPTIMAL MU-SCo CONSTRUCTIONS

The DE-SCo construction in the previous section simultaneously satisfies two receivers with parameters (B_1, T_1) and $B_2 = \alpha B_1$, and $T_2^* = \alpha T_1 + B_1$ for any integer $\alpha > 1$ with a rate $R = \frac{T_1}{T_1 + B_1}$. From the single user capacity bound, clearly this is the maximum possible rate for any MU-SCo code. Furthermore *DE-SCo* is also optimal for any other receiver with $T_2 > T_2^*$. Below we state the regimes in which the optimal MU-SCo constructions are known [9].

Theorem 2. Suppose that the burst-delay parameters for user 1 and 2 are (B_1, T_1) and (B_2, T_2) respectively. Suppose that $B_2 = \alpha B_1$ for some integer $\alpha > 1$. Then optimal MU-SCo constructions are known for the following values of T_2 :

$$C = \begin{cases} \frac{T_1}{T_1 + B_1}, & T_2 \geq T_2^* \triangleq \alpha T_1 + B_1, \\ \frac{T_2 - B_1}{T_2 - B_1 + B_2}, & \max\{B_2, T_1\} + B_1 \leq T_2 \leq T_2^*, \\ \frac{T_1}{T_1 + B_2}, & T_1 \leq T_2 \leq T_1 + B_1, B_2 \leq T_1 \\ \frac{T_2}{T_2 + B_2}, & T_2 \leq T_1 \end{cases} \quad (6)$$

The different regions in (6) are also summarized in Fig. 3. For clarity we ignore the integer proportionality constraints on the parameters in this figure (extension for non-integer values is included in [9]). The first case in (6) corresponds to regions marked (a) and (b). In region (a), the capacity can be achieved by a simpler interference avoidance construction (illustrated in Table. II(a)) rather than the DES-Co construction. Region (c) corresponds to case 2 in (6). The capacity is achieved through a DES-Co construction for a modified value of T_1 . The example of $\{(1, 2) - (2, 4)\}$ in Section III falls in this region. The full derivation is provided in [9]. Region (f) corresponds to case 3 in (6). The capacity is achieved by a (B_2, T_1) SCo code and finally for region (g), which corresponds to the last case in (6), the capacity is achieved by a (B_2, T_2) SCo code. In regions (d) and (e) the capacity remains open except for a special case

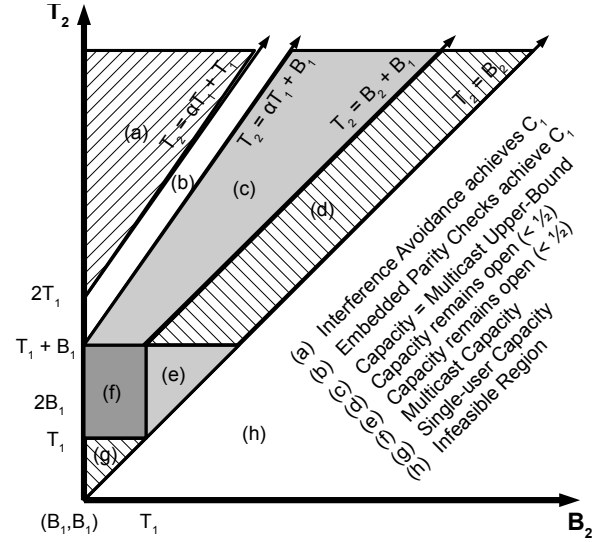


Fig. 3: Achievable Capacity Regions. Suppose B_1 and T_1 are constants, so the regions depend on the relation between T_2 and B_2 .

when $T_2 = B_2$ and $T_1 = B_1$ where the concatenation based scheme is known to be optimal [3].

VI. CONCLUSION

This paper constructs a new class of streaming erasure codes that do not commit apriori to a given delay, but rather achieve a delay based on the channel conditions. We model this setup as a multicast problem to two receivers whose channels introduce different erasure-burst lengths and require different delays. The DE-SCo construction embeds new parity checks into the single-user code, in a such a way that the weaker receiver can also recover the stream with an information theoretically optimum delay. MU-SCo constructions that are optimal for a range of burst-delay parameters are also provided.

At a broader level, our paper sheds insight into the interesting impact delay can have in multiuser channels. How can one order two receivers — one with a weaker channel but a relaxed delay constraint and one with a stronger channel but a stringent delay requirement? The present paper illustrates the intricate interaction between delay and channel quality. We hope that this paper sparks further interest into this fascinating area.

REFERENCES

- [1] E. Martinian, "Dynamic Information and Constraints in Source and Channel Coding," *Ph.D. Thesis*, MIT, September 2004.
- [2] E. Martinian and M. Trott, "Delay-optimal Burst Erasure Code Construction," *ISIT*, (Nice, France) July 2007.
- [3] A. Khisti and J.P. Singh, "On multicasting with streaming burst-erasure codes," *ISIT*, (Seoul, Korea) July 2009.
- [4] F. Jelinek, "Upper bounds on sequential decoding performance parameters," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 227–239, Mar. 1974.
- [5] A. Sahai, "Why Do Block Length and Delay Behave Differently if Feedback Is Present?," *IEEE Transactions on Information Theory*, May 2008, Vol. 54, pp. 1860 — 1886.
- [6] J.K.Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in *IEEE ISIT 2008*, (Toronto, Canada), Jul. 2008.
- [7] C. Chang, S. C. Draper and A. Sahai, "Lossless coding for distributed streaming sources," submitted *IEEE Trans. Inform. Theory*, 2007.

- [8] M.Kalman, E.Steinbach,and B.Girod, "Adaptive media playout for low-delay video streaming over error-prone channels," IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, pp. 841—851, 2004.
- [9] A. Badr, A. Khisti, and E. Martinian, "Diversity Embedded Streaming Erasure Codes (DE-SCo): Constructions and Optimality", In preparation.